

DOCUMENT RESUME

ED 388 220

IR 017 369

AUTHOR Shute, Valerie J.
TITLE Regarding the I in ITS: Student Modeling.
PUB DATE 94
NOTE 9p.; In: Educational Multimedia and Hypermedia, 1994.
Proceedings of ED-MEDIA 94--World Conference on
Educational Multimedia and Hypermedia (Vancouver,
British Columbia, Canada, June 25-30, 1994); see IR
017 359.
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference
Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Intelligent Tutoring
Systems; Knowledge Representation; Models; Thinking
Skills
IDENTIFIERS Conceptual Frameworks

ABSTRACT

For an intelligent tutoring system (ITS) to earn its "I", it must be able to (1) accurately diagnose students' knowledge structures, skills, and/or learning styles using principles, rather than pre-programmed responses, to decide what to do next; and (2) adapt instruction accordingly. While some maintain that remediation actually comprises the "T" in ITS, this paper takes the position that the two components (diagnosis and remediation), working in concert, make up the intelligence in an ITS. A framework for developing and assessing student models is presented, followed by a description of an attempt to apply the framework in the development of a student model incorporated within a non-intelligent computer tutor. The two systems (with and without a student model) are compared in terms of outcome and efficiency measures. The framework is an adaptation of Dillenbourg and Self's (1992) two-dimensional framework and notation for student modeling, which was modified to represent specific knowledge and skill types required during the learning process, procedural skills, conceptual knowledge rather than overt behaviors, and cognitive process measures. The horizontal axis remains basically the same as the original: learner's representation of the knowledge or the skill, system's representation of the learner's knowledge, and the system's/expert's representation of the knowledge or skill. This modified framework represents the standard microadaptive approach to student modeling. The intelligent and non-intelligent version of "Stat Lady," an experiential learning environment and curriculum that teaches statistical concepts and skills, are described. The "Stat Lady" versions provide the basis for a planned experiment testing the degree to which inclusion of a student model may enhance learning outcome measures and/or improve learning efficiency. (Contains 16 references.) (Author/MAS)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

Regarding the *I* in ITS: Student Modeling

VALERIE J. SHUTE
Armstrong Laboratory
Brooks Air Force Base, Texas 78235-5352 USA
shute@eis.brooks.af.mil

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

C This document has been reproduced as
received from the person or organization
originating it
C Minor changes have been made to improve
reproduction quality

• Points of view or opinions stated in this docu-
ment do not necessarily represent official
OERI position or policy

Abstract: For an intelligent tutoring system (ITS) to earn its "I", it must be able to (a) accurately diagnose students' knowledge structures, skills, and/or learning styles using principles, rather than pre-programmed responses, to decide what to do next, and then (b) adapt instruction accordingly. While some maintain that remediation actually comprises the "T" in intelligent tutoring systems, my position is that the two components (diagnosis and remediation), working in concert, make up the intelligence in an ITS. A framework for developing and assessing student models is presented, followed by a description of an attempt to apply the framework in the development of a student model incorporated within a non-intelligent computer tutor. The two systems (with and without a student model) will be compared in terms of learning outcome and efficiency measures.

Jeremy (age 10) arrives at his math lab where he sits in front of a computer that is going to help him learn to solve algebra word problems. Today's focus is on those troublesome distance-rate-time problems. After stating his name, the computer accesses Jeremy's records, flagging his relevant strengths and weaknesses (i.e., not only his higher-level aptitudes from his computerized school records, but also the low-level rules that he's acquired and not yet acquired in this module). Beginning with an animated review of concepts and skills that he learned the day before, the computer generates a problem which is just a little bit beyond his grasp. The system then works out the correct solution to the problem, along with some alternative solutions that Jeremy is likely to come up with based on its student model of him. In fact, he incorrectly solves the problem like the tutor predicted. As part of its student model of him, the computer "knows" to instruct Jeremy with an emphasis on a graphical representation of the problem to clarify the discrepancy between the correct and incorrect solutions and facilitate the formation of a functional mental model (conceptual knowledge). Thus, the tutor presents two animated trains appearing on opposite sides of the screen, and converging at a point almost in the middle of the screen. They travel at different rates of speed. The problem statement stays up at the top of the screen, and the tutor points out, as it periodically pauses the simulation, what elements should be attended to and when. After Jeremy states that he understands the mapping between the explicated conceptual knowledge, the appropriate equation, and the relevant parts of the word problem, the computer presents an isomorphic word problem. This time he solves it correctly, without any supplemental graphics. The computer allows him to play around with some trains, missiles and boats on his own for a while to test his emerging understanding. He views his "score" of curricular elements acquired, and instruction and learning continue.

The above scenario could describe events in a class-lab 15 to 50 years from now, or just remain a figment of our imaginations. To achieve this future, we need to conduct more systematic research on student modeling, focusing on increased system flexibility and diagnostic accuracy. We also need to adopt some framework and formalism for a more precise specification of models. All this presumes that the student model is the correct focus for developing more intelligent tutoring systems, so we additionally need controlled evaluations testing which modeling techniques are better for which kinds of domains, and even whether or not a student model, in general, is worth the research and development costs.

The ability to diagnose student errors and tailor remediation based on the diagnosis represents the critical difference between intelligent and merely clever computer-assisted instruction. The working definition of computer-tutor intelligence that I'll be using in this paper is that a system must *behave* intelligently, not actually *be* intelligent, like a human. Specifically, an intelligent system must be able to accurately diagnose students'

2017369
"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

Gary H. Marks

knowledge structures, skills, and/or learning styles using principles, rather than pre-programmed responses, to decide what to do next, and then adapt instruction accordingly (Shute & Psotka, in press).

Generic Intelligent Tutoring System

A student learns from an intelligent tutoring system (ITS) primarily by solving problems--ones that are appropriately selected or tailor-made, and that serve as good learning experiences for that student. The system starts by assessing what the student already knows. This is called the *student model*. The system concurrently must consider what the student needs to learn, embodied in the curriculum (or *domain expert*) as instructional goals. Finally, the system must decide what curriculum element (unit of instruction) ought to be instructed next, and how it shall be presented. This is achieved by the inherent teaching strategy, or *tutor*, in communication with the student model. From all of these considerations, the system selects or generates a problem, then either works out a solution to the problem (via the domain expert), or retrieves a prepared solution. The tutor then compares its solution, in real-time, to the one the student has prepared and performs a diagnosis based on differences between the two. Feedback is offered by the ITS based on issues such as how long it's been since feedback was last provided, whether the student already received some particular advice, what the student's strengths and weaknesses are, and so on. After this, the program updates the student model (the dynamic record of what the student knows and doesn't know). Following these updating activities, the entire cycle is repeated, starting with selecting or generating a new problem.

Not all ITS include these components, and the problem-test-feedback cycle does not adequately characterize all systems. However, this generic depiction does describe many current ITS. Alternative implementations exist, representing philosophical as well as practical differences in their design. For example, the standard approach to building a student model involves representing emerging knowledge and skills of the learner. The computer responds to updated observations with a modified curriculum that is minutely adjusted. This may be called a *microadaptive* approach to modeling, where instruction is very much dependent on individual response histories. An alternative, *macroadaptive* approach involves assessing incoming knowledge and skills, either instead of, or in addition to, emerging knowledge and skills. This enables the curriculum to adapt to both persistent and/or momentary performance information as well as their interaction (see Shute, 1993a, 1993b). I'll now present a student modeling framework designed to aid in the development and comparison of different modeling techniques by providing a standard formalism.

A Framework for Student Modeling

Dillenbourg and Self (1992) outlined a two-dimensional framework and notation for student modeling. Their vertical dimension distinguishes among learner behavior, behavioral knowledge, and conceptual knowledge. This is crossed by the second dimension reflecting the representation of that knowledge--by the learner, the system, and the system's representation of the learner. I modified the framework slightly (see Figure 1) to represent specific knowledge and skill types required during the learning process (i.e., symbolic knowledge, SK; procedural skill, PS; and conceptual knowledge, CK), rather than overt behaviors. The horizontal axis remains basically the same (i.e., learner's representation of the knowledge or skill, L; the system's representation of the learner's knowledge, S_L; and the system's/expert's representation of the knowledge or skill, S). This modified framework reflects the standard microadaptive approach to student modeling.

I also included in the framework a fourth element on the vertical axis reflecting cognitive process measures (or aptitudes). This was added to accommodate individual differences among learners (for macroadaptation), as well as differential cognitive requirements of problems or tasks. Finally, although not shown in the figure, I incorporated a correlated time dimension because, as learning progresses, cognitive demands on the individual vary. For example, working-memory capacity and associative learning skills play an important role early in the learning process in determining the degree of new knowledge and skill acquisition. But over time, these become less important, and other cognitive factors (e.g., perceptual-motor speed) gain importance.

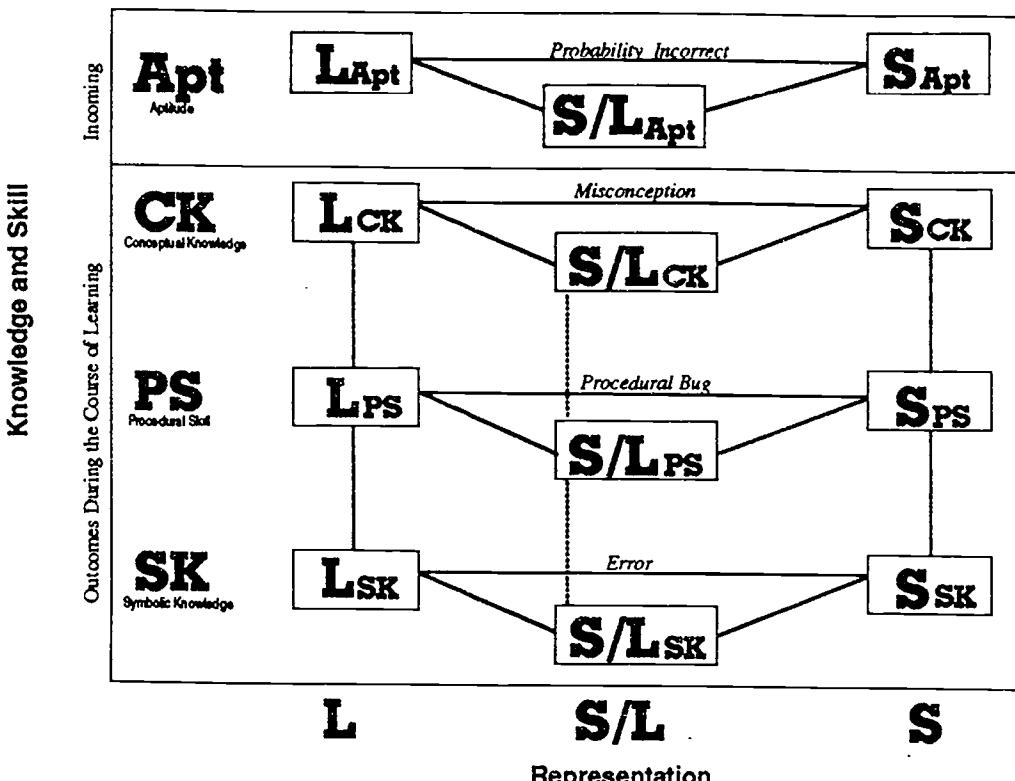


Figure 1: Framework for Student Modeling (adapted from Dillenbourg & Self, 1992)

Let's see how this works. Suppose you were developing a computer program to teach statistical topics, and you were at the part of the curriculum instructing the computation of a measure of central tendency—the MEAN. Your system begins by introducing relevant symbolic knowledge (e.g., Σ , X, N, $\Sigma X/N$), then requiring learners to solve problems to demonstrate their acquisition of this new knowledge. Differences between the learner's and the system's representation (or understanding) of the knowledge would show up as *errors* on specific problems, such as failing to recognize the denominator (N) as being a part of the final formula (see bottom section of Figure 1, above). Next, the system directs students to apply their new knowledge by actually computing the MEAN from a set of data (procedural skill). Disparities between the learner's actual solution process (and product) and the system's representation of the correct procedure would be apparent in *procedural bugs*. Finally, learners are taught (and induce), the conceptual knowledge that the MEAN represents the arithmetic average of a set of data, and how that relates to other measures of central tendency (e.g., the location of the MEAN in relation to the MEDIAN and MODE in a skewed distribution). Discrepancies between the learner's and system's conceptual representations (or mental models) can reflect fundamental *misperceptions*.

Another variable believed to influence the probability that a learner will get a specific problem correct or incorrect involves the match between a learner's aptitude (L_{Apt}) and the cognitive requirements of a specific task (S_{Apt}). That is, if the working memory demands/requirements for task X were high (e.g., introduction of a new curricular element requiring the integration of diverse knowledge), and the learner's actual working-memory capacity was low, the predicted probability of success would be low. Cognitive process measures (aptitudes) can easily be assessed. For example, the Cognitive Abilities Measurement (CAM) battery of computerized tests (Kyllonen, Woltz, Christal, Shute, Tirre, and Chaiken, 1990) measures six different aptitudes (i.e., working-memory capacity, inductive reasoning skills, information processing speed, associative learning skills, procedural learning skills, and general knowledge) in each of three domains (quantitative, verbal, and spatial). The battery has been widely-tested and validated, and individual tests can be extracted for different purposes. For instance, if you were teaching a three-dimensional navigational task and wanted to know an individual's spatial aptitude, you could select certain tests for administration (e.g., tests measuring spatial working memory). Differences between the learners actual aptitude (L_{Apt}) and the system's representation of the learner's aptitude (S/L_{Apt}) denotes the construct validity of the aptitude measure (e.g., CAM test validity). Moreover, discrepancies between the system's

representation of the learner's aptitude (S/L_{Apt}) and the system's aptitude requirements (S_{Apt}) indicate what kind of remediation is required (e.g., decomposing the current task into more manageable units for a learner diagnosed as having low working-memory capacity).

This general framework also allows for extensions. For instance, the learner's representation of his/her own knowledge (e.g., L/L_{CK}) would denote metacognitive awareness of his or her underlying knowledge structure. I'll now use this framework as a basis for discussing student modeling within a single tutor that attempts to teach a variety of knowledge types to different learners.

A Real-World Attempt to Add Intelligence to a Computer Tutor

Non-Intelligent Version of Stat Lady

Stat Lady (SL) consists of an experiential learning environment and a curriculum that teaches statistical concepts and skills (e.g., DESCRIPTIVE STATISTICS). The curriculum is presented in a relatively fixed order to all learners. The curriculum focused on in this paper was built on a hierarchy of simple to more complex concepts and skills, and consists of three 2-hour modules: (a) data organization and plotting, (b) measures of central tendency, and (c) measures of variability. In addition to student problem solving within the three modules, and the provision of specific feedback, the system also allows learners to engage in elective "extracurricular" activities, such as viewing the on-line Dictionary and Formula Bank, playing around in the Number Factory, or using the Grab-a-Graph features.

The non-intelligent version of SL is often clever, but not intelligent in the classical sense (see working definition, above). The type of pedagogy embedded in the system may be called "near mastery learning." That is, relevant concepts and rules are presented, then SL poses various problems for students to solve in order to demonstrate comprehension of the curricular elements. If a learner continues to get a problem wrong, SL provides three levels of feedback, progressively more explicit. For any particular problem, if the student fails to solve it correctly after three attempts, he or she is given the correct answer. The system thus *presumes* that the learner has actually acquired that concept or skill after explicitly being told. But this could be an erroneous presumption, with problems arising later on when a student tries to learn higher-level skills that have only partially-learned subskills as components. *As the twig is bent, so grows the tree.*

Suppose that two students, Tonya and Nancy, have advanced to the "Measures of Central Tendency" module. Specifically, they are beginning the section concerning the computation of the MEAN. *Stat Lady* begins this topic by identifying the MEAN as the most commonly used measure of Central Tendency, and as being synonymous with the *average*--a term familiar to most students. After presenting these broad conceptual strokes, *Stat Lady* quickly illustrates the concepts with a real-life example (e.g., computing an average grade in Biology class from a group of test scores).

Tonya and Nancy then must obtain some test scores from the Number Factory (an activity that's considered analogous to data collection), and *Stat Lady* walks them through the process of building the formula for the MEAN. Even though they're only in the earliest stages of learning this new concept, *Stat Lady* constantly challenges her students to think for themselves by drawing on bits and pieces of relevant information covered in a previous section of the tutor. For instance, rather than simply giving Tonya and Nancy the formula for the MEAN, she tests their symbolic knowledge (SK) and associative learning abilities by asking them to use a menu of options to create the notation that stands for summing scores in a sample (ΣX). They'd worked with these two symbols before, but never conjointly. Right away, we see some individual differences in learning--Nancy has no problem with the task just described. She enters the correct answer on her first attempt, is congratulated with positive auditory and textual feedback, and moves directly on to actually summing her scores (i.e., a test of procedural skill, PS). Tonya, on the other hand, is struggling. She enters a series of incorrect responses, each of which is followed by the increasingly explicit feedback (and encouragement) mentioned above. *Stat Lady* finally gives Tonya the answer and tells her to go back and enter it, and then Tonya is advanced to the "Computation of the Sum" exercise.

While both students eventually proceed through the identification of "N" as the denominator of the formula, determination of the actual sample size, and the computation of the MEAN of the obtained test scores, an examination of their performance histories reveals two very different stories. Nancy is successful time and again, and even computes the MEAN correctly on her first attempt. *Stat Lady* congratulates her and she moves on to solve

colorful problem sets, applying the formula she's just created. In contrast, Tonya has a rough time of it. She never seems to catch on and has to be given the answer to everything except the actual sample size (N), a concept she remembered from the "Data Organization and Plotting" module.

Even a quick illustration such as this makes the inherent problem apparent. Although it seems safe to assume that Nancy has a firm understanding of the MEAN (symbolic knowledge and procedural skill), the same can't be said for poor Tonya. It's *possible* that, at some point during, or after, the presentation of the full formula, all of the puzzle pieces coalesce for Tonya and she actually acquires the desired knowledge and skill, but there's no firm *proof* that this ever happened. Ideally, all students would walk away from a tutor having demonstrated the knowledge level that Nancy evidenced. How can *Stat Lady* be modified to yield more guarantees of successful knowledge and skill acquisition?

Intelligent Version of Stat Lady

The first step in rendering *Stat Lady* intelligent was to perform an extensive cognitive task analysis of the curriculum, employing two subject-matter experts for reliability. This yielded more than 250 curricular elements that were classified into: symbolic knowledge, procedural skills, and conceptual knowledge. Next, these elements were arranged in a hierarchy, relating higher-level knowledge and skills to one another and to successively lower-level knowledge and skills. In the example above concerning the acquisition of symbolic knowledge of the MEAN, correctly specifying the formula presumes familiarity with (if not complete comprehension of) four lower-level SK curricular elements: the summation notation (Σ), variable values (X), total sample size (N), and the way in which these components are combined ($\Sigma X/N$). A more *conceptual* understanding of this concept requires comprehension of frequency distributions, learned earlier, as well as realizing that the MEAN represents one of three measures of central tendency, all related to one another. Moving up the hierarchy, the MEAN also relates to issues of variability, where computing or understanding standard deviation is dependent on successful acquisition of the MEAN (SK, PS, and CK). Unfortunately, the size limitations of this paper make it impossible to depict the entire network.

The intelligent version of *Stat Lady* (ISL) includes a student model. Actually, there are three student models, one each for SK, PS, and CK, but I'll just refer to them collectively as "student model" because they apply the same basic updating heuristics. The student model is represented by a directed graph with values associated with each node/element showing the mastery level of each student. The edges of the graph represent prerequisite knowledge and skills. This is called an "inheritance hierarchy" where different elements are shown along with their "children" or component parts.

Initial values for the model are obtained from students' performance on a comprehensive pretest designed to assess incoming knowledge of *all* curriculum elements resulting from the cognitive task analysis. Based on the assessment of pretest performance, subjects are then placed into the appropriate part of the curriculum (macroadaptation). For instance, if a learner correctly answered all items relating to frequency distributions, proportions, percentages, MEAN, MEDIAN, and MODE, those items would all receive a "1" in the preliminary student model. If he or she failed all items relating to variance, standard deviation, skewness, and kurtosis, those values would retain their initialized values of "0" and that individual would start the tutor at the section of the curriculum containing the lowest level of curricular element containing a value of "0."

Fuzzy student modeling variables may be associated with each curricular element. Values assigned to these variables range from 0 to 1 representing knowledge states, from "most likely unknown" to "most likely known" (see Derry & Hawkes, 1993; Lesgold, Eggan, Katz, & Rao, 1992). The preliminary student model, instantiated with binary (0/1) data from the pretest performance, is then continuously updated during the course of learning--at the conclusion of each problem solving endeavor. Updating (i.e., upgrading and downgrading values) occurs first at the lowest levels (e.g., individual components), then is propagated up the hierarchy. A student is moved on to the next appropriate unit of instruction upon exceeding a student-model value of .89, the mastery criterion for each curricular element.

Following placement in the curriculum from pretest assessment, *Stat Lady* introduces the current topic, and students proceed to solve related problems. The student can correctly solve the problem on his/her first try, or incorrectly solve it. As described above, SL has three levels of feedback for incorrect responses, thus a person can receive 0, 1, 2, or 3 hints for a given problem. The probability of successful acquisition of that problem is simply defined as: $[1 - (\text{misses}/3)] * \text{Pretest constant}$ (see Figure 2). The Pretest constant is: if correct, multiply by .91; if

incorrect, multiply by .89. Learners therefore won't be advanced if they are not quite successful (< .9). These two differential weights (.91, .89) are used because, if a learner correctly solves an item on the pretest and is also correct on her first attempt during problem solution, the evidence suggests that she knows the item and can be advanced given the above-threshold value ($[1 - 0/3] * .91 = .91$). This is slightly higher than the case where a person misses a particular item on the pretest, but solves the problem correctly without any hints ($[1 - 0/3] * .89 = .89$) during the tutor. In the latter case, the learner does not advance to the next unit of instruction because it's easier to solve a problem following some instruction compared to initially coming up with an answer on one's own, as required in the pretest. Beginning values for the eight different conditions are shown below.

		Problem Performance Number of Misses (Number of Hints)			
		0	1	2	3
Pretest Performance	1	.91	.61	.30	0
	0	.89	.59	.30	0

Figure 2: Student Model of One Curricular Element Following a Problem-Solving Episode

For values less than .9, the system presents a fresh problem, and, depending on the learner's performance, the new value is combined with the old to yield a composite success score. To illustrate, suppose Nancy failed all pretest items relating to curriculum element X. When placed in the curriculum to learn about X, she required one hint before successfully solving the problem. Her initial student model value would be: .59 (i.e., $(1 - 1/3) * .89$). Another problem is presented, and this time she gets it correct without any assistance, so her new score is 1 ($1 - 0/3 = 1$) (Note: the pretest function only enters into the equation with the first problem solving episode). The combined success score at this point updates to: $(1 + .59)/2 = .8$. While close, this value is still slightly below criterion performance. Thus a third problem is presented, and Nancy completes the problem effortlessly (with no hints). The combined score now reaches threshold $(.8 + 1)/2 = .9$, and Nancy begins the next topic. In contrast, a learner beginning with a value of 0 would require a minimum of 4 fresh problems before advancement, providing he or she was successful on each of those (i.e., start at 0, then update to .5, .75, .88, .94). Slower students may take even more problems, carefully tailored to fit the specifically-diagnosed deficits or misconceptions.

These values are also propagated up the inheritance hierarchy to provide preliminary estimates for complex curricular elements that contain element X as a component knowledge or skill. As mentioned, the STANDARD DEVIATION requires an understanding of the MEAN in addition to an understanding about distributions, deviations from the MEAN, sum of squares, and so forth. The current student-model value for MEAN, then, becomes part of the preliminary value for the standard deviation, along with other information, such as pretest performance data, complexity of the curricular element, and additional information concerning the learner's aptitude (the last two have not yet been implemented). A curricular element's difficulty or complexity may be based on indicators such as the number of required elements and operators, expert ratings of difficulty, and empirically-determined indices of element difficulty. The learner's aptitude may be assessed via the CAM battery, mentioned earlier.

The student model is visible to the student in the form of an on-line Report Card, presented as a bar graph showing the level of mastery the system believes the student has achieved (i.e., S/L_{SK} , S/L_{PS} , S/L_{CK}). The Report Card represents concepts and skills at a global level (e.g., SK of the MEAN, PS in computing the MEAN), then is further decomposable into individual elements (e.g., success on Σ , N, X, $\Sigma X/N$). Additional records are maintained on students' usage of the elective tools, referred to earlier (e.g., number of times they accessed the on-line Dictionary). This information is inspectable as well, and eventually will work its way into the formal student model. That is, a person who supplements his/her tutorial instruction by engaging in self-initiated reading of the

hypertext Dictionary, for instance, is probably more successful in knowledge and skill acquisition (compared to passive learners) given their active, motivated learning behaviors.

Discussion

There is a large cost associated with incorporating a student model into a tutoring system. This raises two important questions: (1) How much, and what kind of, information about a learner is required to adequately diagnose knowledge and skill acquisition and subsequently tailor instruction to the learner's needs? (2) What is the payoff of increasing a system's adaptability? Sleeman (1987) has argued that "... if one takes seriously the findings of the ATI work of Cronbach and Snow (1977), it would appear that there is little likelihood of producing instruction that is uniquely individualized" (p. 242). The key word in this statement is "uniquely." An exhaustive characterization of a learner would probably not warrant the effort and expense in terms of increases in final outcome. But the empirical question remains: How much is enough? In other words, does inclusion of a student model in a tutor enhance overall learning outcome or efficiency? There is equivocal evidence in the literature concerning these issues. In some cases, researchers have reported no advantage of error remediation in relation to learning outcome (e.g., Bunderson & Olsen, 1983; Sleeman, Kelly, Martinak, Ward & Moore, 1989), whereas in others, some advantage has been reported for more personalized remediation (e.g., Anderson, 1993; Shute, 1993a, 1993b; Swan, 1983).

Another question asks whether the student model is even the right framework around which to build good learning systems. Derry and Lajoie (1993) presented several reasons why the student modeling paradigm is problematic. Among the more compelling reasons cited were that: (a) In complex domains, the student model can not specify all possible solution paths, nor determine all possible "buggy" behaviors, (b) Reflection and diagnosis should be performed by the student, not the tutor, and (c) Model-tracing is only applicable to procedural learning, but the focus should be on critical thinking and problem solving. The approach to student modeling taken in this paper does not attempt to delineate all possible solution paths or buggy behaviors. Rather, the information about curriculum elements, derived from the cognitive task analysis and arranged in an inheritance hierarchy, provides a basis for inferences about what knowledge and skills have been acquired and to what degree. Furthermore, this approach models not only procedural skill acquisition, but also symbolic and conceptual knowledge acquisition.

Another contribution of this approach is the inclusion of aptitude in the equation as a predictor of subsequent learning. That is, ATI research provides information about initial states of learners that can be applied in *macroadaptive* instruction (e.g., selection of Jeremy's graphics emphasis in the opening scenario, or location in the curriculum that a learner should be placed to maximize instructional efficiency). Subsequently, *microadaptive* instruction can be used as a response to particular actions (e.g., selection of the next small unit of instruction to be presented based on a specific response history). Initial states may be characterized by an aptitude profile, then microadaptive instructional systems can focus on strengths, circumvent weaknesses, or highlight deficits to be remediated.

The two versions of *Stat Lady*, described herein, provide the basis for an experiment testing the degree to which inclusion of a student model may (a) enhance learning outcome measures, and/or (b) improve learning efficiency. Approximately 400 subjects will be run in an upcoming experiment (August, 1994) testing the efficacy of this student modeling approach. Subjects will be randomly assigned to one of the two versions of *Stat Lady* (intelligent vs. non-intelligent). An on-line pretest will be administered, students will proceed through their respective tutors, and then a posttest will be given. We will be looking at pretest to posttest changes in scores, learning rates (the systems are both self-paced), as well as any aptitude by treatment interactions. Finally, this study will provide the basis for a cost-benefit analysis and some preliminary answers to burning questions about the worth of student modeling.

References

- Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.
Bunderson, C. V., & Olsen, J. B. (1983). *Mental errors in arithmetic skills: Their diagnosis in precollege students* (Final project report, NSF SED 80-125000). Provo, UT: WICAT Education Institution.
Cronbach, L. J., & Snow, R. E. (1977). *Aptitudes and instructional methods: A handbook for research on interactions*. New York: Irvington.

- Derry, S. J., & Lajoie, S. P. (1993). A middle camp for (un)intelligent instructional computing: An introduction. In S. P. Lajoie & S. J. Derry (Eds.), Computers as cognitive tools (pp. 1-11). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Dillenbourg, P., & Self, J. (1992). A framework for learner modelling. Interactive Learning Environments, 2(2), 111-137.
- Derry, S. J., & Hawkes, L. W. (1993). Local cognitive modeling of problem-solving behavior: An application of fuzzy theory. In S. P. Lajoie & S. J. Derry (Eds.), Computers as cognitive tools (pp. 107-140). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Kyllonen, P. C., Woltz, D. J., Christal, R. E., Shute, V. J., Tirre, W. C., & Chaiken, S. (1990). CAM-4: Computerized battery of cognitive ability tests. Unpublished computer program, Brooks Air Force Base, TX.
- Lesgold, A., Eggan, G., Katz, S., & Rao, G. (1992). Possibilities for assessment using computer-based apprenticeship environments. In J. W. Regian & V. J. Shute (Eds.), Cognitive approaches to automated instruction (pp. 49-80). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Self, J. A. (1993). Model-based cognitive diagnosis. User-Modeling and User-Adapted Interaction, 3, 89-106.
- Shute, V. J. (1993a). A macroadaptive approach to tutoring. Journal of Artificial Intelligence and Education, 4(1), 61-93.
- Shute, V. J. (1993b). A comparison of learning environments: All that glitters... In S. P. Lajoie & S. J. Derry (Eds.), Computers as Cognitive Tools (pp. 47-74). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Shute, V. J., & Psotka, J. (in press). Intelligent tutoring systems: Past, present, and future. To appear in D. Jonassen (Ed.), Handbook of Research on Educational Communications and Technology. Scholastic Publications.
- Sleeman, D. H. (1987). PIXIE: a shell for developing intelligent tutoring systems. In R. Lawler & M. Yazdani (Eds.), AI and education: Learning environments and intelligent tutoring systems (Vol. 1, pp. 239-265). Norwood, NJ: Ablex Publishing.
- Sleeman, D. H., Kelly, A. E., Martinak, R., Ward, R. D., & Moore, J. L. (1989). Studies of diagnosis and remediation with high school algebra students. Cognitive Science, 13(4), 551-568.
- Snow, R. E. (1990). Toward assessment of cognitive and conative structures in learning. Educational Researcher, 18(9), 8-14.
- Swan, M. B. (1983). Teaching decimal place value: A comparative study of conflict and positively-only approaches (Research Rep. No. 31). Nottingham, England: University of Nottingham, Sheel Center for Mathematical Education.

Acknowledgements

I would like to sincerely thank several people who have been working with me on this project: Roger Dubbs, Kevin Gluck, Lee Gugerty, Pat Kyllonen, Nancy Lefort, Dave Logan, and Wes Regian. The research reported in this paper was conducted by personnel of the Armstrong Laboratory, Human Resources Directorate, Brooks Air Force Base, Texas. The opinions expressed in this article are those of the author and do not necessarily reflect those of the Air Force.

Correspondence concerning this paper should be addressed to Valerie J. Shute, AL/HRMI, 7909 Lindbergh Drive, Brooks AFB, Texas 78235-5352.